



# lab 8

```
#include <iostream>
#include <vector>
using namespace std;

class Person {
private:
    string name;
    int age;
    string ssNum;

public:
    Person() {}
    Person(string name, int age, string ssNum) {
        this->name = name;
        this->age = age;
        this->ssNum = ssNum;
    }
    string getName() { return name; }
    int getAge() { return age; }
    string getSsNum() { return ssNum; }
    void setName(string name) { this->name = name; }
    void setAge(int age) { this->age = age; }
    void setSsNum(string ssNum) { this->ssNum = ssNum; }
};

class spouse : public Person {
private:
    string anniversary;

public:
    spouse() {}
    spouse(string name, int age, string ssNum, string anniversary)
        : Person(name, age, ssNum) {
        this->anniversary = anniversary;
    }
    string getAnniversary() { return anniversary; }
    void setAnniversary(string anniversary) { this->anniversary = anniversary; }
};

class child : public Person {
private:
    string favoriteToy;

public:
    child() {}
    child(string favoriteToy, string name, int age, string ssNum) {
        this->favoriteToy = favoriteToy;
    }
};
```

```

        this->setName(name);
        this->setAge(age);
        this->setSsNum(ssNum);
    }
    string getFavoriteToy() { return favoriteToy; }
    void setFavoriteToy(string favoriteToy) { this->favoriteToy = favoriteToy; };
};

class division {
    string divisionName;

public:
    division() {}
    division(string divisionName) { this->divisionName = divisionName; }
    string getDivisionName() { return divisionName; }
    void setDivisionName(string divisionName) {
        this->divisionName = divisionName;
    }
};

class jobDescription {
    string description;

public:
    jobDescription() {}
    jobDescription(string description) { this->description = description; }
    string getDescription() { return description; }
    void setDescription(string description) { this->description = description; }
};

class Employee : public Person {
private:
    string employeeNum;
    string hireDate;
    string title;
    vector<child> children;
    spouse *spouse;
    division division;
    vector<jobDescription> jobDescriptions;

public:
    Employee() {}
    Employee(string name, int age, string ssNum, string employeeNum,
             string hireDate, string title, vector<child> children,
             class spouse *spouse, class division division,
             vector<jobDescription> jobDescriptions)
        : Person(name, age, ssNum) {
        this->employeeNum = employeeNum;
        this->hireDate = hireDate;
        this->title = title;
        this->children = children;
        this->spouse = spouse;
        this->division = division;
        this->jobDescriptions = jobDescriptions;
    }
    string getEmployeeNum() { return employeeNum; }
    string getHireDate() { return hireDate; }
    string getTitle() { return title; }
};

```

```

vector<child> getChildren() { return children; }
class spouse *getSpouse() {
    return spouse;
}
class division getDivision() {
    return division;
}
vector<jobDescription> getJobDescriptions() { return jobDescriptions; }
void setEmployeeNum(string employeeNum) { this->employeeNum = employeeNum; }
void setHireDate(string hireDate) { this->hireDate = hireDate; }
void setTitle(string title) { this->title = title; }
void setSpouse(class spouse *spouse) { this->spouse = spouse; }
void setDivision(class division division) { this->division = division; }
void setJobDescriptions(string description) {
    jobDescription jobDescription(description);
    jobDescriptions.push_back(jobDescription);
}
void setChild(class child child) { children.push_back(child); }
};

void printEmployee(class Employee Employee) {
    cout << Employee.getName() << endl;
    cout << Employee.getAge() << endl;
    cout << Employee.getSsNum() << endl;
    cout << Employee.getEmployeeNum() << endl;
    cout << Employee.getHireDate() << endl;
    cout << Employee.getTitle() << endl;
    for (int i = 0; i < Employee.getChildren().size(); i++) {
        cout << "child " << i << ": " << Employee.getChildren()[i].getName()
            << endl;
        cout << "child " << i << ": " << Employee.getChildren()[i].getAge() << endl;
        cout << "child " << i << ": " << Employee.getChildren()[i].getSsNum()
            << endl;
        cout << "child " << i << ": " << Employee.getChildren()[i].getFavoriteToy()
            << endl;
    }
    cout << Employee.getSpouse()->getName() << endl;
    cout << Employee.getSpouse()->getAge() << endl;
    cout << Employee.getSpouse()->getSsNum() << endl;
    cout << Employee.getSpouse()->getAnniversary() << endl;
    cout << Employee.getDivision().getDivisionName() << endl;
    for (int i = 0; i < Employee.getJobDescriptions().size(); i++) {
        cout << "job description " << i << ": "
            << Employee.getJobDescriptions()[i].getDescription() << endl;
    }
}

int main() {
    Employee bat("bat", 20, "123", "Employee1", "2010/10/10", "Boss",
        vector<child>(), nullptr, division("123"),
        vector<jobDescription>());
    bat.setChild(child("toy", "child1", 1, "123"));
    bat.setChild(child("toy", "child2", 2, "123"));
    bat.setSpouse(new spouse("spouse", 20, "123", "2010/10/10"));
    bat.setJobDescriptions("job1");
    bat.setJobDescriptions("job2");
    printEmployee(bat);
}

```

```
    return 0;  
}
```